

---

# **Eigenfish Documentation**

***Release 1.0***

**Seth Pendergrass, Zhe Bai and Steven Brunton**

**Oct 16, 2016**



<b>1</b>	<b>eigenfish package</b>	<b>3</b>
1.1	Subpackages	3
1.1.1	eigenfish.classify package	3
	Submodules	3
	eigenfish.classify.classify module	3
	Module contents	4
1.1.2	eigenfish.process package	4
	Submodules	4
	eigenfish.process.math module	4
	eigenfish.process.process module	4
	Module contents	4
1.2	Submodules	4
1.3	eigenfish.eigenfish module	4
1.4	eigenfish.util module	5
1.5	Module contents	5
<b>2</b>	<b>Indices and tables</b>	<b>7</b>
	<b>Python Module Index</b>	<b>9</b>
	<b>Index</b>	<b>11</b>



Contents:



---

## eigenfish package

---

## 1.1 Subpackages

### 1.1.1 eigenfish.classify package

#### Submodules

#### eigenfish.classify.classify module

**class** eigenfish.classify.classify.**Classifier**

Bases: object

**classify** (*data*)

Classifies data based on current model.

**Parameters** **data** – Matrix with each column a different sample.

**Returns** List of predictions, where return[i] describes data[:, i].

**cross\_validate** (*data, labels*)

Cross-validates trained model against data with labels.

**Parameters**

- **data** – Matrix with each column a different sample.
- **labels** – List of labels, each corresponding to a column of data.

**Returns** Percent labels the same.

**load** (*filename*)

Loads trained model from file, overwriting current model. Do not use on training files you did not create.

**Parameters** **filename** – Name of file to load.

**save** (*filename*)

Saves trained model to filename.

**Parameters** **filename** – Name of file to save model as.

**train** (*data, labels*)

Trains current classifier with matrix data and labels, where labels[i] describes data[:, i].

**Parameters**

- **data** – Matrix of data, where each column is a separate sample.

- **labels** – List of labels, each corresponding to a column of data.

## Module contents

### 1.1.2 eigenfish.process package

#### Submodules

##### eigenfish.process.math module

`eigenfish.process.math.fft2_series` (*img\_mat*, *shape*)

For each column in *img\_mat*, *img\_mat[:, i]* the fft2 modes are extracted and placed into the corresponding column of the returned matrix.

#### Parameters

- **img\_mat** – Matrix to process.
- **shape** – Original (width, height) of each column of *img\_mat*.

**Returns** New `numpy.ndarray` matrix, where *return[:, i]* is the fft2 modes of *img\_mat[:, i]*.

`eigenfish.process.math.rpca` (*image\_mat*)

Performs Robust Principle Component Analysis on *image\_mat*.

**Returns** Low-rank, sparse parts of *image\_mat*

##### eigenfish.process.process module

**class** `eigenfish.process.process.Processor`

Bases: `object`

**process** (*img\_mat*, *shape*)

Process *img\_mat* to prepare it for training/classification.

#### Parameters

- **img\_mat** – Matrix with each column a flattened image.
- **shape** – Original (width, height) of each image.

## Module contents

### 1.2 Submodules

### 1.3 eigenfish.eigenfish module

**class** `eigenfish.eigenfish.Eigenfish` (*shape*, *training\_file=None*, *processor=None*, *classifier=None*)

Bases: `object`

**classify** (*img\_mat*)

Classify *img\_mat* based on current training.

**Parameters** **img\_mat** – Column-wise matrix of flattened images.



**Returns** List of labels, one for each column of `img_mat`.

**cross\_validate** (*img\_mat*, *label\_arr*)

Cross-validates the trained model. `img_mat` will be run through the classifier, and each predicted label of `img_mat[:, i]` compared with `label_arr[i]`. The percent same is returned.

**Parameters**

- **img\_mat** – Column-wise matrix of flattened images.
- **label\_arr** – List of labels, where `label_arr[i]` corresponds to `img_mat[:, i]`.

**Returns** Percent of labels that are the same.

**load** (*filename*)

Loads saved training data and overwrites current model. Use only on data you have previously saved, and make sure to use the same processor and classifier.

**Parameters filename** – File to load into classifier.

**save** (*filename*)

Saves currently trained model to filename.

**Parameters filename** – File to save from classifier.

**train** (*img\_mat*, *label\_arr*)

Add to current model's training.

**Parameters**

- **img\_mat** – Column-wise matrix of flattened images.
- **label\_arr** – List of labels, where `label_arr[i]` corresponds to `img_mat[:, i]`.

## 1.4 eigenfish.util module

`eigenfish.util.load_img_mat` (*files*)

Loads all files as images in black and white, flattens them and returns them as a `numpy.ndarray`.

**Parameters files** – List of image files to load. All should be of the same resolution.

**Returns** `Numpy.ndarray` matrix with each column a flattened images.

## 1.5 Module contents



---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`



## e

- `eigenfish`, 5
- `eigenfish.classify`, 4
- `eigenfish.classify.classify`, 3
- `eigenfish.eigenfish`, 4
- `eigenfish.process`, 4
- `eigenfish.process.math`, 4
- `eigenfish.process.process`, 4
- `eigenfish.util`, 5



## C

Classifier (class in eigenfish.classify.classify), 3  
classify() (eigenfish.classify.classify.Classifier method), 3  
classify() (eigenfish.eigenfish.Eigenfish method), 4  
cross\_validate() (eigenfish.classify.classify.Classifier method), 3  
cross\_validate() (eigenfish.eigenfish.Eigenfish method), 5

## E

Eigenfish (class in eigenfish.eigenfish), 4  
eigenfish (module), 5  
eigenfish.classify (module), 4  
eigenfish.classify.classify (module), 3  
eigenfish.eigenfish (module), 4  
eigenfish.process (module), 4  
eigenfish.process.math (module), 4  
eigenfish.process.process (module), 4  
eigenfish.util (module), 5

## F

fft2\_series() (in module eigenfish.process.math), 4

## L

load() (eigenfish.classify.classify.Classifier method), 3  
load() (eigenfish.eigenfish.Eigenfish method), 5  
load\_img\_mat() (in module eigenfish.util), 5

## P

process() (eigenfish.process.process.Processor method), 4  
Processor (class in eigenfish.process.process), 4

## R

rpca() (in module eigenfish.process.math), 4

## S

save() (eigenfish.classify.classify.Classifier method), 3  
save() (eigenfish.eigenfish.Eigenfish method), 5

## T

train() (eigenfish.classify.classify.Classifier method), 3  
train() (eigenfish.eigenfish.Eigenfish method), 5